## REMARKS/ARGUMENTS

Claims 1, 4-8, 12, 15-18, and 21-24 are pending in the present application. No claims were added or canceled; claims 1, 6, 12, 17-18, and 23-24 were amended. Reconsideration of the claims is respectfully requested.

### I. Objection to Claims: Claims 1, 6, 12, 17-18, and 23-24

The Office Action has stated that claims 1, 6, 12, 17-18, and 23-24 were objected to for certain informalities. Specifically, the Office Action objected to claims 1, 6, 12, 17-18, and 23-24 for the use of the term "one of ... or ...."

In rejecting the claims, the Examiner states:

> **Claims 1, 6, 12, 17-18, and 23-24** are objected for the use of the term "one of...**or**..." The Examiner suggests Applicant amend to the term to instead read "one of...**and**..." as does appear in various limitations of claims 1, 12, 18, and 24. Appropriate correction required.

Office Action dated August 24, 2007, page 3 (emphasis in original).

In response, Applicants have amended claims 1, 6, 12, 17-18, and 23-24 to recite "and" as required by the Office Action.

### II. 35 U.S.C. § 103, Obviousness: Claims 1, 4-8, 12, 15-18, and 21-24

The Office Action has rejected claims 1, 4-8, 12, 15-18, and 21-24 under 35 U.S.C. § 103(a) as being unpatentable over *Damron*, System and Method for Prefetching for Pointer Linked Data Structures, U.S. Patent No. 6,782,454, August 24, 2004 (hereinafter "*Damron*") in view of *Hooker,* Microprocessor with Repeat Prefetch Instruction, U.S. Patent Application Publication No. 2003/0191900, October 9, 2003 (hereinafter "*Hooker*"). This rejection is respectfully traversed.

In rejecting the claims, the Examiner states:

> **As per claims 1 and 18,** Damron discloses a method in a data processing system for providing hardware assistance to prefetch data during execution of code by a processor in the data processing system, the method comprising:
>
> responsive to loading of an instruction in the code into a cache, determining, by a processor unit, whether metadata for a prefetch is present for the instruction, wherein the metadata comprises a prefetch indicator that is associated with the instruction by one of being placed in the instruction or stored in a shadow memory (col. 4, lines 58-61; col. 5, lines 17-24; Fig. 1, element 175; Fig. 3, element 220); *It should be noted that computer program product in claims* 18-23 *executes the exact same functions as the methods in claims I and 4-6. Therefore, any references that teach claims* 1-6 *also teach the corresponding claims 18 and* 2 1-23. *It should also be noted that the "prefetch request"*

> *is analogous to the "instruction", the "prefetch engine" is analogous to the "processing unit': and the "starting address of a node (to be prefetched), an offset value, and a termination value" all in combination are analogous to the "metadata." Lastly, it should be noted that the "starting address of a node (to be prefetched)" and the "offset value", which are placed in the prefetch request, are analogous to the "prefetch indicator" being placed in the instruction.*
>
> responsive to determination of the metadata being present for the instruction, determining whether data is to be prefetched (col. 5, lines 27-35; Fig. 3, element 230); *It should be noted that "determining whether a termination condition has been satisfied" is analogous to "determining whether data is to be prefetched." Data is prefetched until the termination condition is met.*
>
> and responsive to a determination that data is to,be prefetched, prefetching data, from within a data structure using the metadata, into the cache in the processor (col. 5, lines 24-26 and 35-42; Fig. 3, elements 225 and 235). *It should be noted that responsive to the termination condition not being met, data is prefetched.*
>
> Damron does not expressly disclose wherein the step of determining whether data is to be prefetched comprises one of determining whether a number of outstanding cache misses is less than a threshold, and determining whether a number of cache lines chosen to be replaced is greater than athreshold;
>
> and wherein the step of prefetching comprises one of prefetching the data responsive to determining that the number of outstanding cache misses is less than the threshold, and prefetching the data responsive to determining that the number of cache lines chosen to be replaced is greater than the threshold.
>
> Hooker discloses wherein the step of determining whether data is to be prefetched comprises one of determining whether a number of outstanding cache misses is less than a threshold, and determining whether a number of cache lines chosen to be replaced is greater than a threshold (paragraph 0069; Fig. 5, element 536); I*t should be noted that the "rtesponse buffers" are analogous to the "cache lines."*
>
> and wherein the step of prefetching comprises one of prefetching the data responsive to determining that the number of outstanding cache misses is less than the threshold, and prefetching the data responsive to determining that the number of cache lines chosen to be replaced is greater than the threshold (paragraph 0070; Fig. 5, element 538).
>
> Damron and Hooker are analogous art because they are from the same field of endeavor, that being prefetching memory systems.
>
> At the time of the invention it would have been obvious to a person of ordinary skill in the art to implement Hooker's threshold based prefetch method within Damron's prefetch engine because, advantageously, by not prefetching cache lines if not enough free response buffers exist, the efficiency of a microprocessor is potentially increased.
>
> Therefore, it would have been obvious to combine Damron and Hooker for the benefit of obtaining the invention as specified in claims 1 and 18.

Office Action dated August 24, 2007, pages 3-6 (emphasis in original).

The Examiner bears the burden of establishing a *prima facie* case of obviousness based on prior art when rejecting claims under 35 U.S.C. § 103. *In re Fritch*, 972 F.2d 1260, 23 U.S.P.Q.2d 1780 (Fed. Cir. 1992). The prior art reference (or references when combined) must teach or suggest all the claim limitations. *In re Royka,* 490 F.2d 981, 180 USPQ 580 (CCPA 1974). In determining obviousness, the scope and content of the prior art are... determined; differences between the prior art and the claims at issue are... ascertained; and the level of ordinary skill in the pertinent art resolved. Against this background the obviousness or non-obviousness of the subject matter is determined. *Graham v. John Deere Co.*, 383 U.S. 1 (1966). "Often, it will be necessary for a court to look to interrelated teachings of multiple patents; the effects of demands known to the design community or present in the marketplace; and the background knowledge possessed by a person having ordinary skill in the art, all in order to determine whether there was an apparent reason to combine the known elements in the fashion claimed by the patent at issue." *KSR Int'l. Co. v. Teleflex, Inc.*, No. 04-1350 (U.S. Apr. 30, 2007). "*Rejections on obviousness grounds cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness.* Id. (citing *In re Kahn*, 441 F.3d 977, 988 (CA Fed. 2006))."

Amended independent claim 1, which is representative of amended independent claims 12, 18, and 24 with regards to similarly recited subject matter, recites:

> 1. A method in a data processing system for providing hardware assistance to prefetch data during execution of code by a processor in the data processing system, the method comprising:
>
> responsive to loading of an instruction in the code into a cache, determining, by a processor unit, whether metadata for a prefetch is present for the instruction, wherein the metadata comprises a prefetch indicator that is associated with the instruction by one of being placed in the instruction and stored in a shadow memory;
>
> responsive to a determination of the metadata being present for the instruction, determining whether data is to be prefetched, wherein the step of determining whether data is to be prefetched comprises one of determining whether a number of outstanding cache misses is less than a first threshold value, and determining whether a number of cache lines chosen to be replaced is greater than a second threshold value; and
>
> responsive to a determination that data is to be prefetched, prefetching data, from within a data structure using the metadata, into the cache in the processor, wherein the step of prefetching comprises one of prefetching the data responsive to determining that the number of outstanding cache misses is less than the first threshold value, and prefetching the data responsive to determining that the number of cache lines chosen to be replaced is greater than the second threshold value.

The combination of *Damron* in view of *Hooker* fails to render obvious the present invention as the combination of *Damron* in view of *Hooker* fails to teach the presently claimed invention. More specifically, the combination of *Damron* in view of *Hooker* fails to teach the features of **wherein the step of determining whether data is to be prefetched comprises one of determining whether a number of outstanding cache misses is less than a first threshold value, and determining whether a number of cache lines chosen to be replaced is greater than a second threshold value;** and **wherein the step of prefetching comprises one of prefetching the data responsive to determining that the number of outstanding cache misses is less than the first threshold value, and prefetching the data responsive to determining that the number of cache lines chosen to be replaced is greater than the second**

**threshold value**, as recited in claim 1.

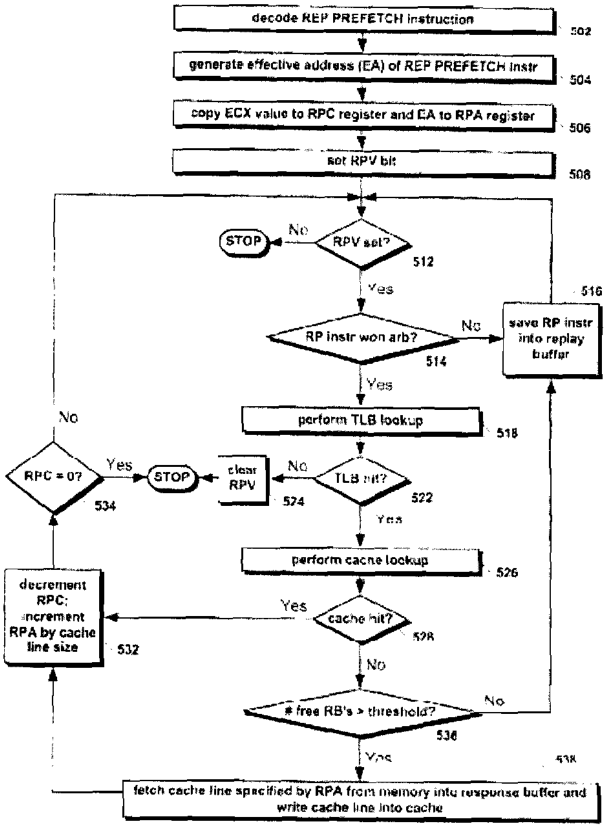
On pages 4 and 5 of the Office Action mailed August 24, 2007, the Office Action admits, and Applicants agree, that *Damron* does not teach the features of“wherein the step of determining whether data is to be prefetched comprises one of determining whether a number of outstanding cache misses is less than a first threshold value, and determining whether a number of cache lines chosen to be replaced is greater than a second threshold value; and wherein the step of prefetching comprises one of prefetching the data responsive to determining that the number of outstanding cache misses is less than the first threshold value, and prefetching the data responsive to determining that the number of cache lines chosen to be replaced is greater than the second threshold value." However, the Office Action cites to *Hooker* as allegedly teaching these features.

Specifically, the Office Action points to *Hooker*, paragraph [0069], Figure 5 element 536, which is reproduced below for the Examiner's convenience, as allegedly teaching the feature of wherein the step of determining whether data is to be prefetched comprises one of determining whether a number of outstanding cache misses is less than a first threshold value, and determining whether a number of cache lines chosen to be replaced is greater than a second threshold value; and paragraph [0070], Figure 5 element 538, which is also reproduced below for the Examiner's convenience, as allegedly teaching the feature of wherein the step of prefetching comprises one of prefetching the data responsive to determining that the number of outstanding cache misses is less than the first threshold value, and prefetching the data responsive to determining that the number of cache lines chosen to be replaced is greater than the second threshold value:

> [0068] At decision block 534, control logic 144 determines whether the RPC 188 has a value of 0 to determine whether all the cache lines specified by the repeat prefetch instruction 400 have been prefetched. If not, flow proceeds back to decision block 512 to prefetch another cache line. Otherwise, flow ends.
>
> [0069] At decision block 536, control logic 144 determines whether result signal 192 of FIG. 1 generated by comparator 178 of FIG. 1 indicates the number of free response buffers 166 of FIG. 1 is greater than the value stored in threshold register 182 of FIG. 1. If not, flow proceeds to block 516. Otherwise, flow proceeds to block 538. In order to prefetch a cache line, control logic 144 must allocate one of the response buffers 166 to the prefetch, which may result in subsequently starving a higher priority instruction from allocating a response buffer 166 it needs. Advantageously, by not prefetching cache lines if not enough free response buffers 166 exist, the efficiency of the microprocessor 100 is potentially increased.
>
> [0070] At block 538, all the conditions have been satisfied for the repeat prefetch instruction 400 to prefetch the next cache line. Consequently, control logic 144 allocates a response buffer 166 and instructs the bus interface unit 172 of FIG. 1 to fetch the cache line specified by the RPA 186 from system memory into the allocated response buffer 166. When the bus interface unit 172 fetches the cache line into the response buffer 166, the cache line is then written into the cache 154 according to the locality hint specified in

the ModR/M byte 406 of FIG. 4. Flow proceeds from block 538 to block 532 to increment the RPA 186 and decrement the RPC 188 and prefetch the next cache line if necessary and if all the conditions are met.

FIG. 5


decode REP PREFETCH instruction
502
generate effective address (EA) of REP PREFETCH instr
504
copy ECX value to RPC register and EA to RPA register
506
set RPV bit
508
STOP
No
RPV set?
512
Yes
RP instr won arb?
514
No
save RP instr into replay buffer
516
Yes
No
perform TLB lookup
518
RPC = 0?
534
Yes
STOP
clear RPV
524
No
TLB hit?
522
Yes
perform cache lookup
526
decrement RPC; increment RPA by cache line size
532
Yes
cache hit?
528
No
# free RB's > threshold?
536
No
Yes
538
fetch cache line specified by RPA from memory into response buffer and write cache line into cache

Repeat Prefetch Instruction Operation

*Hooker*, paragraphs [0068]-[0070], Figure 5.

Paragraph [0069] of *Hooker*, reproduced above, fails to teach or suggest the feature of wherein the step of determining whether data is to be prefetched comprises one of determining whether a number of outstanding cache misses is less than a first threshold value, and determining whether a number of cache lines chosen to be replaced is greater than a second threshold value. Paragraph [0069] of *Hooker* teaches that a free response buffer is allocated to a prefetch only if the total number of free response buffers is above a certain threshold value. This is done to insure that there are enough free response buffers to be assigned to handle higher priority instructions. This is, however, completely different from what is recited in claim 1. Claim 1 recites determining whether a number of cache lines chosen to be replaced is greater than a second threshold value. *Hooker*, in paragraph [0069], fails to teach or suggest either of these features.

*Hooker* teaches **determining if a number of response buffers available to be allocated is above a threshold value,** wherein the threshold value represents a number of response buffers to be held in reserve. In contradistinction, claim 1 recites **determining if a number of cache lines to be replaced**

**is below a first threshold value.** ***Hooker*** **does not teach, suggest, or even mention determining a number of cache lines to be replaced.** Further, *Hooker* teaches determining if the number of available response buffers is above a threshold, whereas, claim 1 recites determining whether the number of cache lines to be replaced is above a second threshold value. Thus, ***Hooker*** **teaches determining** if there is at least a certain **amount of freespace available,** whereas **claim 1 teaches determining** whether **the number of cache lines are to be replaced** is above a threshold value.

Additionally, paragraph [0069] of *Hooker* fails to teach, suggest, or even mention **determining whether a number of outstanding cache misses is less than a first threshold value.** Additionally, no other portion of *Hooker* teaches the feature **determining whether a number of outstanding cache misses is less than a first threshold value.** Paragraph [0068] of *Hooker*, reproduced above, teaches determining if RPC 188 has a value of zero. If RPC has a value of zero, then all prefetch instructions have been prefetched. If the value is not zero, then the next cache line is simply prefetched. Paragraph [0044] of *Hooker* defines RPC 188 as a repeat prefetch count, which is output from a repeat prefetch count register. *Hooker* does not teach or suggest determining whether a number of outstanding cache misses is less than a threshold. Rather, *Hooker* teaches prefetching a cache line as long as an unfulfilled prefetch instruction exists. That is, *Hooker* teaches prefetching cache lines as long the number of unfulfilled prefetch request is above a value of zero. In contradistinction, claim 1 recites **determining whether a number of outstanding cache misses is less than a first threshold value.** Thus, *Hooker* fails to teach or suggest **determining whether a number of outstanding cache misses is less than a threshold.**

Futhermore, paragraph [0068] of ***Hooker*** **teaches prefetching a cache line as long as a prefetch instruction exists.** In contradistinction, **claim 1 recites the step of determining whether data is to be prefetched comprises one of determining whether a number of outstanding cache misses is less than a first threshold value, and determining whether a number of cache lines chosen to be replaced is greater than a second threshold value.** Therefore, *Hooker* fails to teach or suggest the feature of **wherein the step of determining whether a number of outstanding cache misses is less than a first threshold value, and determining whether a number of cache lines chosen to be replaced is greater than a second threshold value**, as recited in the claims of the present invention.

As *Hooker* fails to teach or suggest the feature of **wherein the step of determining whether data is to be prefetched comprises one of determining whether a number of outstanding cache misses is less than a first threshold value, and determining whether a number of cache lines chosen to be replaced is greater than a second threshold value**, logically, *Hooker* also fails to teach the feature of "wherein the step of prefetching comprises one of prefetching the data **responsive to determining that the number of outstanding cache misses is less than the first threshold value,** and prefetching the data

**responsive to determining that the number of cache lines chosen to be replaced is greater than the second threshold value.**" The Office Action cites to *Hooker*, paragraph [0070], reproduced above as allegedly teaching this feature. However, paragraph [0070] of *Hooker* merely teaches that if the conditions set out in paragraph [0069] are met, prefetching the cache line into the allocated buffer and decrementing the RPC to indicate that one prefetch instruction has been satisfied. Paragraph [0070] of *Hooker*, does not teach or suggest the feature of **wherein the step of prefetching comprises one of prefetching the data responsive to determining that the number of outstanding cache misses is less than the first threshold value, and prefetching the data responsive to determining that the number of cache lines chosen to be replaced is greater than the second threshold value**, as recited in claims 1 of the present invention.

Therefore, for at least the reasons set forth above, Applicants submit that the combination of *Damron* in view of *Hooker* fails to render claim 1 obvious. Applicants respectfully submit that the Office Action fails to state a *prima facie* case of obviousness in regards to claim 1. Thus, Applicants submit that claim 1 is in condition for allowance over the cited references. Furthermore, as claim 1 is representative of independent claims 12, 18, and 24, the same distinction between claim 1 and the cited references also apply to claims 12, 18, and 24. Additionally, as claims 4-8, 15-17, and 21-23 depend from independent claims 1, 12, and 18, Applicants submit that claims 4-8, 15-17, and 21-23 are also in condition for allowance at least by virtue of their depending from an allowable base claim.

Therefore, the rejection of claims 1, 4-8, 12, 15-18, and 21-24 under 35 U.S.C. § 103(a) has been overcome.

## III. Conclusion

It is respectfully urged that the subject application is patentable over the combination of *Damron* in view of *Hooker* and is now in condition for allowance.

The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: November 20, 2007

Respectfully submitted,

/Gerald H. Glanzman/
Gerald H. Glanzman
Reg. No. 25,035
Yee & Associates, P.C.
P.O. Box 802333
Dallas, TX 75380
(972) 385-8777
Attorney for Applicants

GHG/blj